
Red de apoyo local



UNIVERSIDAD COMPLUTENSE MADRID

Autores

Adrián CÓRDOBA CLAPÉS

Carlos AMBRONA PALACIOS

Carlos PERDOMO FERNÁNDEZ

Marta HUERTAS SMOLIS

Yelim KIM

Director

Carlos GREGORIO RODRÍGUEZ

31 de mayo de 2019

Autores

Adrián Córdoba Clapés
Grado en Ingeniería de Computadores

Carlos Ambrona Palacios
Grado en Ingeniería de Computadores

Carlos Manuel Perdomo Fernández
Grado en Ingeniería de Computadores

Marta Huertas Smolis
Grado en Ingeniería de Software

Yelim Kim
Grado en Ingeniería de Software

May 31, 2019

Resumen

Las redes sociales suelen estar constituidas por los miembros de la familia nuclear, amigos y amigas, vecinos y vecinas, compañeros y compañeras de trabajo y personas conocidas en la comunidad. Estos vínculos se convierten en sistemas de apoyo social, los cuales constituyen un elemento indispensable para la salud, ajuste y bienestar del individuo. El aspecto importante de las redes de apoyo es que son personas emocionalmente significativas para la persona en el caso que nos concierne y son quienes potencialmente brindan el apoyo social. Es decir estas redes sociales cumplen la función de proveer apoyo social, el cual genera herramientas necesarias para la persona dentro de su contexto cotidiano.

Teniendo en cuenta esto, las nuevas tecnologías nos brindan una buena herramienta para facilitar a personas con este deseo de buscar compañía de otra persona en situaciones de temor o ansiedad la posibilidad de establecer un proceso de comparación social, a través del cual se puede obtener información acerca de la situación estresante (lo que permite reducir la incertidumbre) y validar las propias reacciones (comparando sentimientos y conductas).

En esta memoria, detallaremos las herramientas que se van a poner a disposición de las usuarias de nuestra red de apoyo local.

Abstract

Social networks are usually made up of members of the nuclear family, friends, neighbours, coworkers and acquaintances of the community. These links become social support systems, which are an essential element for the health, adjustment and well-being of the individual. The important aspect of support networks is that they are emotionally significant persons for the individual in the case that concerns us and are those who potentially provide social support. In other words, these social networks fulfill the function of providing social support, which generates necessary tools for the person within their daily context.

Taking into account the above-mentioned point, Telegram and more specifically, the Telegram bots provide us with a good tool to facilitate people with this desire to seek the company of another person in situations of fear or anxiety the possibility of establishing a process of social comparison, through which you can obtain information about the situation stressful (which reduces uncertainty) and validate one's own reactions (comparing feelings and behaviors).

In this report, we will detail the tools that will be made available to the user.

– **Palabras clave**

- Bot
- Telegram
- Python
- Red
- Mujeres
- Ayuda
- Violencia de género
- Mensajería

– **Keywords**

- Bot
- Telegram
- Python
- NetWork
- Women
- Help
- Gender violence
- Messenger

Índice general

Índice general	6
Índice de figuras	8
1 Introducción	9
1.1 Motivación	9
2 Estructuras	11
2.1 Estructura del equipo	11
2.2 Estructura de la memoria	12
3 Metodología	15
3.1 Reunión en la UNED	16
4 Descripción del sistema	19
4.1 Objetivos	19
5 Estudio de las tecnologías	23
5.1 Aplicaciones de mensajería	23
5.2 Lenguajes de programación	28
5.3 Bases de datos	30
5.4 Herramientas de desarrollo	32
6 Arquitectura del sistema	35
6.1 Estructura de BBDD	35
7 Desarrollo del sistema	37
7.1 Implementación del bot	37
7.2 Diagrama de secuencia	39

<i>ÍNDICE GENERAL</i>	7
7.3 Experiencia de usuario - Primer uso	41
8 Puntos críticos del proyecto	51
9 Conclusiones del proyecto	55
10 Ampliaciones futuras	57
10.1 Botón de pánico	57
10.2 Puntuación de usuarios	57
10.3 Ampliación de zonas de chats	58
10.4 Inteligencia artificial	58
11 Aportaciones personales	59
11.1 Adrián Córdoba Clapés	59
11.2 Carlos Ambrona Palacios	61
11.3 Carlos Manuel Perdomo Fernández	63
11.4 Marta Huertas Smolis	65
11.5 Yelim Kim	66
Bibliografía	69

Índice de figuras

2.1	Diagrama de la estructura del equipo	12
3.1	Tablero de organización del equipo	16
7.1	Diagrama de flujo del Bot	40
7.2	Ejemplo de primera conversación con el bot	44
7.3	El mensaje que muestra el bot en el caso de rechazar el	45
7.4	Conversación desde el rol Andrea	46
7.5	Conversación desde el rol red	47
7.6	Diferentes opciones que permite el bot desde el rol Andrea	48
7.7	Diferentes opciones que permite el bot desde el rol Red	49
8.1	Imagen de la herramienta de diseño	51

Capítulo 1

Introducción

En este primer capítulo, se describen nuestras motivaciones a la hora de elegir este proyecto como trabajo de fin de Grado, cómo nos podemos beneficiar de las aplicaciones de mensajería en la actualidad y cómo se está focalizando el mundo de la tecnología en este sector.

La tecnología ha evolucionado de tal manera, que como anunció Luis Miguel Gilpérez (presidente de Telefónica España),

“en el año 2016, el Smartphone se coronó como el principal dispositivo a través del cual los españoles entran en Internet, superando por primera vez al ordenador.” [2]

Esto es debido, en gran medida, al uso de las redes sociales y aplicaciones de mensajería como medio de comunicación. Las tecnologías se adaptan a los usuarios, y ésta en concreto, cada vez integra más servicios.

1.1 Motivación

Como estudiantes de la Universidad Complutense de Madrid (UCM), para la realización de nuestro trabajo final de grado, nos surgió la oportunidad de realizar una idea pensada por profesores y directores de la Universidad Nacional de Educación a Distancia (UNED), con la que estamos totalmente en consonancia, ayudar a mujeres en peligro de violencia machista mediante una red de apoyo local virtual.

Teníamos una idea, la cual debíamos convertir en realidad mediante labores de investigación y los conocimientos adquiridos durante nuestro desarrollo.

Para cumplir con el objetivo, pensamos inmediatamente en los bots que funcionan a través de las apps de mensajería, ya que están totalmente integrados en sus respectivas aplicaciones.

En contra de desarrollar una app propia, ya que hay una cierta fatiga del usuario hacia las apps móviles, la interacción de los usuarios con los bots es más directa que con las apps y son más difíciles de desarrollar y mantener.

Actualmente, existen numerosas aplicaciones de mensajería, las cuales están integrando cada vez más bots conversacionales con su propia IA. La mayoría de la gente tiene alguna de estas aplicaciones instaladas en sus teléfonos (Whatsapp, Telegram, Facebook Messenger).

Aprovechando estas plataformas, hemos decidido desarrollar un bot pensando en ellas, con un fin muy claro: crear una red de apoyo local para mujeres en riesgo de violencia machista.

Estas aplicaciones de mensajería están creando un ecosistema propio gracias a los bots, ya que proporcionan un servicio parecido al de una aplicación pero sin la necesidad de descargarla.

Esta manera ágil de empezar a usar el bot hace que el aumento de una masa crítica necesaria para una red de apoyo local sea más rápida.

Capítulo 2

Estructuras

2.1 Estructura del equipo

Se he elegido la organización Lineo- funcional, la cual consiste en una combinación de Organización Lineal y Funcional.

En una organización lineal las decisiones se concentran en una sola persona, quien toma todas las decisiones y tiene la responsabilidad básica del mando.

La organización funcional, sin embargo, consiste en dividir el trabajo y establecer la especialización de manera que cada miembro tiene su trabajo.

Las ventajas de haber elegido esta estructura son:

- Se obtiene la más alta eficiencia de la persona con mayor especialización.
- La división del trabajo es planeado para evitar los duplicados del trabajo.

La desventaja es la dificultad de localizar y fijar la responsabilidad.

Como cada miembro tenía diferentes habilidades, este tipo de organización es adecuada para maximizar la eficiencia del trabajo de cada persona. Se dividió el trabajo y se decidió cómo avanzar el siguiente paso del proyecto en las reuniones. Tras dichas reuniones de todos los miembros del equipo, se organizaba otra reunión con el profesor para revisar el avance del proyecto. La comunicación constante entre todos los miembros era necesaria porque el trabajo individual de cada persona debía unirse al final con el trabajo de los demás miembros del equipo.

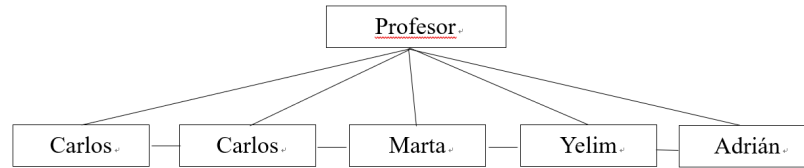


Figura 2.1: Diagrama de la estructura del equipo

2.2 Estructura de la memoria

La memoria de este Trabajo de Fin de Grado consta de la siguiente estructura:

El **Capítulo 1** contiene una breve introducción, además de la motivación que nos ha llevado a realizar este proyecto a lo largo del año. También se describe la estructura del grupo y de la memoria del Trabajo de Fin de grado (esta sección).

En el **Capítulo 3** se explica la metodología de trabajo empleada y el porqué de su elección.

El **Capítulo 4** describe el funcionamiento del sistema, sus aplicaciones y los requisitos. Además se detallará la funcionalidad de la aplicación.

El **Capítulo 5** comenta los estudios de las tecnologías posibles para la implementación del bot y el por qué de la tecnología elegida.

El **Capítulo 6** describe toda la arquitectura del sistema, definiendo los componentes que la forman y explicando cada una de ellas.

El **Capítulo 7** describe la implementación del bot y sus diferentes funcionalidades, así como la experiencia de usuario.

El **Capítulo 8** describe todos los problemas y dificultades que han surgido a lo largo del proyecto. Además, se explica cómo se ha conseguido solucionar dichos problemas y el tiempo que ha conllevado solucionarlos.

El **Capítulo 9** analiza la conclusión del proyecto desde un punto de vista de desarrollo personal y las nuevas capacidades adquiridas gracias al TFG.

El **Capítulo 10** describe las posibles ampliaciones futuras del proyecto.

El **Capítulo 11** describe a título personal, en que apartados del TFG se ha centrado más cada miembro del equipo.

Por último, se encuentran las referencias utilizadas para el desarrollo de esta memoria, información que se ha obtenido para poder explicar los

diferentes componentes utilizados en el proyecto y gracias a ello, poder desarrollar el código de la aplicación.

Capítulo 3

Metodología

Durante el desarrollo de este proyecto se ha aplicado la metodología ágil Scrum.

El método Scrum tiene como objetivo principal mejorar la satisfacción del cliente, la velocidad y calidad de desarrollo a través del trabajo en equipo y la transparencia. Los requerimientos o alcance se cambian durante toda la vida del proyecto.

Se ha considerado que era la metodología más acertada debido a la complejidad del proyecto y a que el nivel de incertidumbre era alto, ya que los requisitos no estaban claramente definidos en un principio.

Para definir los requisitos y encaminar un poco el proyecto se requirieron varias reuniones con nuestros clientes, que eran profesores de la UNED y varios representantes de la asociación contra la violencia de género. Éstos plantearon la necesidad de ayudar de alguna forma a las víctimas y entre todos vimos que la posibilidad de crear una red de apoyo podría ser buena idea además de factible.

Este proyecto, como se ha comentado anteriormente, nació con la idea de crear una red social, pero no estaba claro el funcionamiento ni los límites de la tecnología utilizada.

En el sprint inicial se llevó a cabo una tarea de investigación por parte de todos los miembros, explorando la API de python-telegram-bot para conocer todas las posibilidades y limitaciones de esta librería, ya que el lenguaje escogido para el desarrollo fue Python.

Una vez finalizada la investigación se realizó una reunión con los clientes y se expuso toda la información que se había recabado. Se analizaron los puntos que eran viables, los puntos imposibles y los puntos que podríamos

añadir en un futuro. Gracias a esta reunión se creó el backlog, que consiste en una lista priorizada de requisitos de alto nivel o historias de usuario donde se puede ver en qué tarea se está trabajando actualmente, cuáles se han finalizado y cuáles están pendientes de hacer.

Para la gestión de tareas se ha utilizado Trello como se puede apreciar en la figura 3.1. Se trata de un tablón virtual en el que se pueden colgar ideas, tareas, imágenes o enlaces.

Aplica el sistema Kanban para el registro de actividades con tarjetas virtuales. En la organización de tareas permite agregar listas, adjuntar archivos, etiquetar eventos, agregar comentarios y compartir dichos tableros.

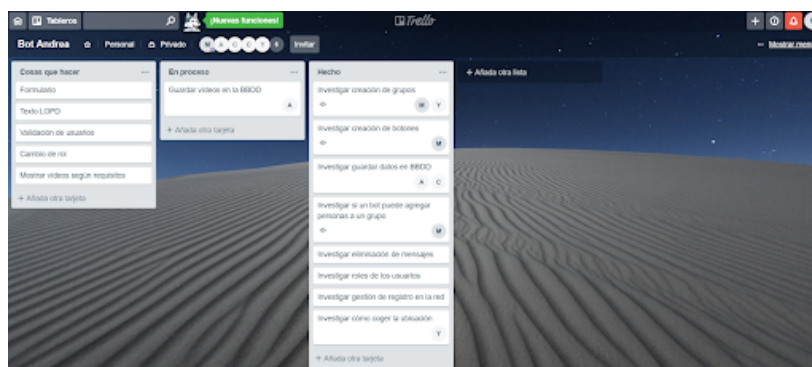


Figura 3.1: Tablero de organización del equipo

Este código está modulado para favorecer futuras mejoras e implementaciones, además lo hemos integrado con una base de datos, de la que posteriormente hablaremos, y que, apoyándonos en el servidor propio de Telegram, soporta vídeos, archivos y ubicación.

Además, para todo esto era interesante el uso de una metodología ágil, lo cual era nuevo para algunos de los miembros, y realizar de manera más eficiente el proyecto.

3.1 Reunión en la UNED

El día 26 de marzo se organizó una reunión con todas las personas involucradas en este proyecto, en la cual se llevó a cabo una demostración delante del grupo promotor y un grupo de consulta (constituido por personas expertas en violencia machista y personas expertas en informática).

Durante la demostración, para la cual se utilizó al web Moqups, se mostraron las posibles funcionalidades que se podrían desarrollar en el bot de Telegram.

En esta reunión se redefinieron los requisitos y se llegó a un acuerdo común sobre las funcionalidades que eran prioritarias, así como un análisis de cómo desarrollarlas dentro de la herramienta.

Se definieron tres metas a alcanzar para nuestra red:

- **Confianza:** personas con las que poder hablar y que ofrezcan apoyo.
- **Acompañamiento:** personas que acompañen en la resolución de problemas tales como trámites, consultas, alternativas, etc. Así como la posibilidad de realizar un acompañamiento físico en tareas diarias como llevar a los niños al colegio, hacer la compra, etc. Se entiende que esto se crea una vez exista una red de confianza.
- **Acción:** para tareas extraordinarias de un enfoque más político, como puede ser organizar una manifestación. Así mismo, esta función estará enfocada a que las personas pasen a la acción involucrándose directamente contra la violencia machista.

Además se debatieron y analizaron los temas relacionados con la seguridad de la víctima, llegando a la conclusión de que sería necesario verificar a todas las personas que entrasen dentro de la red, para lo que se contará con la ayuda de Asociaciones de Violencia de Género. Dichas asociaciones serán las administradoras de la red, cuyo rol se define en el siguiente punto de este documento. La verificación se realizará en dos pasos, el primero mediante el envío de un vídeo al bot y el segundo de forma presencial en alguna de las asociaciones colaboradoras.

Finalmente se definieron los roles de todas las personas que formarán parte de la red, los cuales se explicaran en detalle en el siguiente apartado.

Capítulo 4

Descripción del sistema

En este apartado se describe el funcionamiento del sistema, sus aplicaciones y los requisitos como cliente y servidor. Además, se analiza detalladamente la aplicación.

4.1 Objetivos

Para concretar los requisitos que debía cumplir nuestra red virtual se organizó una reunión los autores de este TFG y el director en conjunto con los organismos de la UNED que pusieron en marcha esta idea, la cual se ha descrito en el punto anterior. A continuación, puntualizamos las funcionalidades principales que se definieron.

Estructura

Para el correcto funcionamiento de la red social se ha definido la siguiente estructura, la cual está dividida en una jerarquía de roles. Dentro de ellos, y descritos más adelante, encontramos: Administradora/s y usuarias. En este último grupo, podemos diferenciar entre ayudantes y ayudadas y localizado dentro de ayudantes, verificadas o sin verificar.

Gestión de usuarios

Para acceder al bot solamente es necesario ser usuario de Telegram, sin embargo se solicitarán ciertos datos por temas de seguridad.

En el caso de ser una Red, tiene que subir un vídeo de presentación corta y una foto del DNI, de esta forma podremos verificar si es una mujer o un hombre, así como la posibilidad de confirmar que no sea una persona peligrosa.

Las personas pertenecientes a la red Verde que quieran subir su categoría, para ser parte de la red Violeta y poder ayudar, tendrán que acudir al centro más cercano indicado por el bot para validar su identificación, de esta forma se comprobará si tiene intención real de ayudar a las víctimas. En cuanto su categoría se actualice, pasarán a pertenecer a la red Violeta automáticamente y tendrán permisos para interactuar con otras personas dentro de la red.

Roles

Tras una serie de preguntas, el bot solicita el rol que desea tener el usuario. Tenemos dos roles principales y existirá la posibilidad de pasar de uno a otro:

- Andrea: se pertenece a la red de voluntarias. Tiene que elegir el género de los vídeos de los ayudantes que le gustaría ver. Se mostrará el vídeo de presentación al azar. Hay tres opciones: Aceptar, Rechazar, y Denunciar el ayudante. Si rechaza o denuncia el ayudante ya no mostrará el vídeo. Si lo acepta, se guarda el vídeo elegido en el registro.
- Red: se solicita ayuda. La red tiene varias capas.
 - Red Verde: Las personas no *certificadas*. Son las que han subido un vídeo de presentación de duración 15 segundos más o menos y una foto del DNI. El bot indicará el centro cercando para que la Red Verde acuda y realizar validaciones.
 - Red Violeta: Las personas *certificadas*. Son las que ya se han validado por los administradores.
 - Red Coral: Las personas *administradoras*. Son las que pueden actualizar las categorías de las Redes, manejar los usuarios denunciados y tener el control en los grupos del chat.

Otras funcionalidades

En la primera interacción con el bot el usuario o usuaria deberá leer un texto con la Ley de Protección de Datos, para poder continuar utilizando el bot deberá pulsar sobre el botón aceptar, en caso contrario el bot dejará de responder.

Tras la aceptación del usuario o usuaria se guardarán sus datos en la base de datos, teniendo la posibilidad de acceder a ellos siempre que lo desee, pudiendo consultarlos o solicitar que se eliminen.

Muchas de las preguntas realizadas por el bot se podrán contestar mediante el uso de botones. En caso de que el usuario use Telegram Desktop, cuando se le solicite la ubicación deberá facilitar su código postal ya que la aplicación no deja mandar ubicaciones. De forma alternativa podrá utilizar Telegram Web.

La ubicación se utilizará para clasificar a los usuarios por zona, de esta forma se les facilitarán los enlaces a los grupos que sean de su zona, donde podrán ayudar a quienes lo soliciten.

Todas las preguntas tendrán la opción de cancelar, así se puede acabar la conversación cuando se desee. Para volver a empezar la conversación el usuario tendrá que introducir el comando [/start](#).

Capítulo 5

Estudio de las tecnologías

En este capítulo se comentan los distintos estudios realizados de las tecnologías disponibles y cuáles son los más apropiados para Andrea.

5.1 Aplicaciones de mensajería

Como se ha comentado anteriormente en esta memoria, se decide utilizar una aplicación de mensajería para la implementación de una red de apoyo ya que son aplicaciones que el gran porcentaje de usuarios de un Smartphone tiene instalada en su móvil, por lo que nos evitamos un primer paso con el que perderíamos muchos usuarios inicialmente. De esta manera el usuario de la red de apoyo, solo tiene que dirigirse al chat de Andrea y comenzar el proceso de registro.

De entre las aplicaciones de mensajería que permiten la programación de bots, las mejores opciones son Facebook Messenger y Telegram:

Facebook Messenger

Facebook Messenger [9] es una aplicación de mensajería que fue lanzada por la red social Facebook, disponible para Android, iOS y Windows Phone. Su gran ventaja es que permite chatear con sus contactos de Facebook.

Esta aplicación también ha decidido apostar por la integración de bot. Además, los bots actúan gracias a la inteligencia artificial de Facebook. Los bots de Facebook Messenger sirven para ayudar a los usuarios a

realizar compras en línea, hacer reservas, jugar a juegos online, consultar información.

KIK Messenger

KIK [11] es una aplicación de mensajería instantánea gratuita para dispositivos móviles, lanzada por la empresa canadiense KIK Interactive. Esta disponible tanto para dispositivos Android como para dispositivos iOS. Uno de sus puntos fuertes es que es totalmente anónima, ya que no requiere de número de teléfono para registrarse.

Al igual que las demás aplicaciones, Kik Messenger también utiliza o permite el uso de bots en su plataforma. Empezaron con bots muy sencillos donde marcas lanzaban ofertas para los usuarios dependiendo del genero, país y dispositivo móvil. Más adelante desarrollaron bots desde el cual podías adquirir comida o productos varios. Por último, añadieron funcionalidad al bot para que pudiera estar en grupos e interactuar con el, así como compartir o reenviar información y links de otros bots a un chat privado o a un grupo.

Whatsapp Messenger (QeuBot)

Whatsapp es una aplicación de mensajería instantánea gratuita multiplataforma, tanto dispositivo móvil como portátil o sobremesa, lanzada en 2009 por un desarrollador ucraniano y posteriormente comprada por Facebook. Su punto fuerte es la gran cantidad de usuarios que la utiliza y su coste gratuito.

Al igual que las demás aplicaciones, Whatsapp permite el uso de un bot en su plataforma. Por ahora las funciones del bot son muy básicas pero abre la puerta a futuros bots con infinidad de posibilidades y funcionalidades. QeuBot [5] tiene entre otras funciones consultar en la Wikipedia, buscar noticias, ejercer de calculadora, buscar imágenes, informar sobre la situación meteorológica o leer el horóscopo.

Telegram

Telegram es una aplicación de mensajería instantánea gratuita, privada y accesible, que fue desarrollada en el año 2013 pero hasta el 2014 no tuvo

una acogida mediática. Desde aquel momento es una de las aplicaciones con mayor número de descargas.

Se puede utilizar en teléfonos, tabletas y ordenadores. Además, se encuentra disponible para más de 10 sistemas operativos: Android, iOS, macOS, Windows, GNU/Linux o navegadores web. Telegram está logrando hacerse su propio hueco gracias a que cada vez se diferencia más de su principal competidor, WhatsApp. Una de las diferencias principales es que Telegram utiliza la arquitectura propia MTProto, con almacenamiento en la nube para los mensajes, bot, canales y supergrupos. El usuario es el que decide si quiere descargar las fotos, vídeos y archivos en el dispositivo para ahorrar memoria o almacenarlos en la nube. Otra característica que le diferencia de WhatsApp es que al ser de código abierto, los desarrolladores pueden crear sus propias aplicaciones dentro de Telegram.

Otro tipo de servicio que ofrece es la mensajería masiva en grupos y en los canales. Se puede llegar a tener un supergrupo con más de 10000 integrantes, los cuales todos los usuarios pueden interactuar entre ellos mandando mensajes de vídeo, animaciones gif, sticker. Los canales son espacios de difusión unilateral para compartir contenido a una cantidad ilimitada de usuarios. Uno de los puntos fuertes de esta aplicación que lo diferencia del resto es el poder desarrollar un bot: es un programa con el que se puede interactuar dentro de la aplicación Telegram como si fuese una persona. Existen los bots que emulan conversaciones normales y los bots interactivos. En estos últimos se pueden tener conversaciones mediante botones o comandos.

Conclusión

Llegamos a la conclusión de que Telegram es la mejor opción en el desarrollo de este proyecto por las siguientes funcionalidades:

- Chat secreto: en esta herramienta tenemos la opción de crear un chat secreto, cuyas ventajas son la posibilidad de activar la autodestrucción de mensajes y la imposibilidad de reenviarlos, lo cual permite que una persona se comuniquen con mayor seguridad y privacidad con otra persona perteneciente a la red. Además estos mensajes van cifrados de punto a punto por una clave generada automáticamente para cada usuario.
- Supergrupos: son grupos en los que se permite hasta 200.000 miembros, se crearán grupos por cada red Andrea que se quiera formar, en la que habrá una persona Andrea y todas las personas de la red violeta y coral que ella desee, pudiendo elegir invitar a más personas o eliminarlas del grupo.
- Eliminación del historial: como en la mayoría de las aplicaciones de chat, Telegram ofrece la posibilidad de eliminar el historial de un grupo o una conversación privada, lo cual puede resultar muy útil si la persona se encuentra en una situación de peligro y no quiere que nadie encuentre esas conversaciones.
- Acceso con contraseña o con biometría: El acceso a la aplicación se puede controlar con un código de 6 dígitos o si el dispositivo lo soporta, mediante huella o identificador facial.
- Archivo de conversaciones: Telegram permite archivar una conversación, es decir, ocultarla del menú de conversaciones, y desarchivarla cuando se desee. Esto permite, en parte, mantener la privacidad por si nuestro dispositivo es utilizado por terceras personas.

Historia de Telegram

La aplicación multiplataforma Telegram es relativamente nueva, ya que se creó en 2013 por los hermanos Nikolai y Pavel Durov en Rusia. Como todo inicio no fue fácil, empezaron 12 desarrolladores a tejer los primeros

hilos de la aplicación que en sus inicios era solo para dispositivos móviles (Android e IOs). En el siguiente año, dieron el salto a multiplataforma y en 2016, aumentaron su núcleo de desarrolladores de 12 personas a 15 personas y 30 ayudantes para políticas de uso.

Desde sus inicios tenían claro la principal ventaja de Telegram, ya que es una seña de identidad de la aplicación y sirve para diferenciarse del resto de las aplicaciones de la competencia, que es su privacidad / seguridad. De hecho, les ha traído muchos problemas y por ello, han tenido que crear nuevas empresas en distintos puntos del mundo para poder librarse de las presiones por parte de gobiernos porque querían tener acceso al contenido de los chat.

Todas las nuevas empresas siempre se rigen bajo una organización apátrida y sin ánimo de lucro, por lo que los ingresos para poder mantener y actualizar la aplicación vienen del sector privado, concretamente de uno de los hermanos fundadores. Aunque en un futuro está previsto que la aplicación sea autosuficiente económicamente hablando.

Por la parte técnica, están en continua actualización y mejora. Desde sus inicios con mensajes de uno a uno y grupales, han ido evolucionando e incluyendo mejoras, como por ejemplo los chat ocultos, compartir ubicación en directo durante un tiempo, enviar vídeos, audios y la posibilidad de mantener conversaciones con mensajes que se eliminan pasado un tiempo tanto en tu dispositivo como en el dispositivo del receptor.

A parte de estar disponible en más de 20 idiomas, tiene versión web, Android, IOs y Windows Mobile.

5.2 Lenguajes de programación

En este apartado se realiza un estudio de los lenguajes de programación que hemos considerado para desarrollar nuestra aplicación dentro de la plataforma Telegram.

Además, se explica cómo se puede desarrollar el bot a partir de los lenguajes propuestos.

Python

Python es un lenguaje de programación gratuito, multiplataforma, de código abierto y fácil de utilizar. Fue creado por Guido van Rossum a finales de los años 80 y en 1991 publicó la primera versión de código 0.9.0. Desde aquel entonces se aceptó como un lenguaje de secuencia de comandos de elección para el geoprocesamiento de usuarios y continúa en crecimiento hasta el día de hoy, con la última versión Python 3.6.2, siendo uno de los lenguajes de programación más empleados para el desarrollo software.

Este lenguaje es fácil de aprender y excelente para principiantes e incluso para expertos. Además, es altamente escalable para grandes proyectos o para pequeños programas. Puede ser utilizado en diversas plataformas y sistemas operativos, como Windows, Mac OS X y Linux, pero también puede funcionar para teléfonos, como por ejemplo Nokia, el cual desarrolló un intérprete para su sistema operativo Symbian.

Empresas y organizaciones bastante conocidas hacen uso intensivo de este lenguaje para desarrollar sus productos y servicios como es el caso de Light and Magic, NASA, Google y Nokia.

Para desarrollar un bot en la plataforma de Telegram, el lenguaje de programación que más se utiliza es éste. Además se puede desarrollar cualquier tipo de software para comunicaciones en red, para aplicaciones de escritorio, para crear juegos y sobre todo para web-apps.

También se puede emplear sus propias librerías para desarrollar el bot, además de conseguir un código mucho más limpio y amigable.

PHP

PHP es un lenguaje de programación de código abierto, es utilizado especialmente para el desarrollo web y puede ser incrustado en HTML.

Apareció en el año 1995 y fue diseñado por Rasmus Lerdof pero la implementación principal de PHP es producida por el The PHP Group. Con el paso de los años ha ido mejorando hasta llegar a la última versión más estable: 7.1.8. Este lenguaje forma parte del software libre publicado bajo la licencia de PHP.

Lo que distingue a PHP es que el código es ejecutado en el servidor, generando un HTML y enviándolo al cliente. Éste recibirá el resultado tras ejecutar el script. El servidor web también puede ser configurado para que procese todos los ficheros HTML con PHP, de tal forma que los usuarios no pueden saber que lleva por debajo.

Al igual que el lenguaje de programación Python, se pueden desarrollar los llamados bots de Telegram sin problema ya que con PHP se pueden hacer llamadas a una dirección HTTP. También se puede utilizar alguna de las librerías de PHP para Telegram Bot de código libre.

Conclusión

La decisión de utilizar un lenguaje de programación fue gracias a nuestro tutor, el cual nos recomendó usar Python ya que es fácil de utilizar para aquellas personas que partan desde cero en este entorno. Cabe destacar que ninguno de los miembros del equipo tenía conocimientos de Python, También nos sugirió utilizar la librería python-telegram-bot. Además, la mayoría de bots que se han visto están desarrollados con apoyo de librería, en parte debido a su amplia documentación y ejemplos. Por todo ello, se decidió elegir el lenguaje de programación Python.

5.3 Bases de datos

En este apartado se realiza un estudio de qué base de datos relacional es la más apropiada para el proyecto desarrollado. Hemos descartado las bases de datos no relacionales, por no tener ninguno de los miembros del equipo experiencia con ellas.

MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos. El lanzamiento fue en 1995. Es considerada como la base de datos *Open Source* más popular del mundo.

Fue escrito en C y C++ y destaca su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación como Python o PHP. Además, se puede integrar en los distintos sistemas operativos.

Es destacable que su utilización sea gratuita e incluso que se pueda modificar con total libertad, pudiendo descargar su código fuente. La principal ventaja que tiene es su gran velocidad al conectar con el servidor como al servir selects y demás. Otra característica es que consume muy pocos recursos, tanto de CPU como de memoria. Además, aunque se cuelgue, no suele perder información y no corrompe los datos.

postgresql

PostgreSQL es un sistema de gestión de base de datos relacional y libre, el cual agiliza la interacción de cliente, servidor y base de datos. Su desarrollo comenzó hace más de 16 años, hasta la última serie de producción que es la 9.1. Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado.

Además, nos proporciona diferentes características como potencia, robustez. Éstas características son las que más se han tenido en cuenta durante su desarrollo.

PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

SQLite

SQLite es una herramienta de software libre que permite almacenar información en dispositivos empotrados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular. SQLite implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo.

Esto permite que SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, y lo más importante es que se puede usar sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al cien por cien entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente.

Otras características de SQLite:

- La base de datos completa se encuentra en un solo archivo.
- Puede funcionar enteramente en memoria, lo que la hace muy rápida. Tiene un footprint menor a 230KB.
- Es totalmente autocontenida (sin dependencias externas).
- Cuenta con librerías de acceso para muchos lenguajes de programación. Soporta texto en formato UTF-8 y UTF-16, así como datos numéricos de 64 bits.
- Soporta funciones SQL definidas por el usuario (UDF). El código fuente es de dominio público y se encuentra muy bien documentado.

Conclusión

Al no conocer en que entorno de producción iba a alojarse Andrea finalmente, nos hemos decantado por SQLite, ya que solo es necesaria una pequeña memoria y una única biblioteca para acceder a bases de datos, lo que lo hace ideal para aplicaciones con esto integrado.

Otro factor a favor de SQLite es que realiza operaciones de manera eficiente y es más rápido que MySQL y PostgreSQL.

También cabe destacar que encajaba perfectamente con Python, ya que dispone de una API sencilla de utilizar.

5.4 Herramientas de desarrollo

En este apartado se van a comentar todas las herramientas de desarrollo utilizadas en el proyecto.

Overleaf

Overleaf [6] es un editor de L^AT_EX online gratuito, previamente llamado ShareLaTeX.

Este editor, al ser online, nos permite tener guardado todo nuestro trabajo en la nube, sin tener la preocupación de pérdida de información. Podemos editar o completar la memoria desde cualquier navegador, con conexión a internet mediante una URL y tener siempre la última versión de la memoria. Incluye un control de versiones, por si en algún momento fuese necesario volver a un punto anterior. Permite que varios usuarios estén modificando el contenido al mismo tiempo y poder ver y compilar los cambios realizados en tiempo real. A parte de tener una opción de descarga de PDF del documento que estas haciendo.

Git

Git es un software de control de versiones de código. Fue creado pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones. Git te brinda un entorno donde alojar tu software y poder ir evolucionándolo en equipo, ya que te ofrece un control de versiones por si en algún momento se detecta algún fallo, poder recuperar una versión anterior. Al ser un repositorio online, puede colaborar mas de un desarrollador y partir todos de la ultima versión. En nuestro caso ha sido muy útil, por la dificultad de cuadrar horarios para reunirnos de forma presencial. Gracias a Git tenemos organizado el código y hemos llevado un control de versiones del código de manera eficiente.

IntelliJ IDEA

IntelliJ [4] es un entorno de desarrollo integrado para el desarrollo de programas informáticos. Tiene compatibilidad tanto para Windows, Mac o Linux. Tiene versión gratuita, la cual es muy completa y nos permite

desarrollar el Trabajo Fin de Grado sin ninguna limitación. Compatible con la mayoría de los lenguajes de programación, entre ellos Python. IntelliJ [10] es muy potente porque tiene conversión de código entre varios lenguajes, a parte de poder trabajar sin problemas con Git, ya que en la parte inferior del interfaz, viene una terminal donde puedes ejecutar los comandos necesarios para actualizarte a la última versión, o con el simple gesto de pinchar en el proyecto con botón derecho y seleccionar la opción Git, te ofrece un gran abanico de posibilidades de comandos como push, pull o resolver conflictos de merge.

Google Drive

Google Drive [3] es un servicio de alojamiento de archivos centralizado en la nube, que permite tener sincronizado todos tus documentos en todo momento desde cualquier dispositivo con acceso a la red. Todo esto nos dio la facilidad de apuntar las ideas o los puntos a tratar que íbamos comentando en las reuniones, de manera que aunque no pudieras asistir a la reunión en tiempo real veías las modificaciones o avances. También guardábamos avances de la memoria, enlaces de interés o errores que nos aparecían al desarrollar código por si a otro compañero le había pasado antes.

Moqups

Bien sea para empezar a crear los primeros bocetos del diseño de una web o hasta para hacer un mockup de una aplicación móvil, siempre es necesario tener a la mano una herramienta de maquetación fácil de usar, minimalista, funcional, completa y preferiblemente gratuita. Eso es moqups que además suma a tales características su funcionamiento en HTML5.

Visual Studio Code

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Es compatible con varios lenguajes de programación [12]. Para el proyecto, era necesario descargar la extensión del Python en el Marketplace.

PyCharm

Es un IDE para programar en el lenguaje Python. Es de la empresa JetBrains que es la misma empresa que la de IntelliJ. PyCharm es compatible con Windows, Mac y Linux. Ya que el trabajo fin de Grado se realizaba en Python era conveniente usar este IDE

Anaconda

Es una distribución libre y abierta de los lenguajes Python y R. Para poder programar en Python, era necesario descargar Anaconda [8] primero y luego asignar el intérprete del código al Python.

Python Telegram Bot API

El Bot API es una interfaz basada en HTTP creada para desarrolladores interesados en crear bots para Telegram. La librería que hemos usado proporciona una interfaz Python pura para la API de Telegram Bot [7]. Además de la implementación de API pura, esta librería presenta una serie de clases de alto nivel para hacer que el desarrollo de bots sea fácil y directo. Estas clases están contenidas en el submódulo telegram.ext. Para poder usar las funciones de esta librería, es imprescindible lanzar el comando `pip install python-telegram-bot==12.0.0b1 -upgrade`

DB Browser for SQLite

Es un sistema de gestión de bases de datos relacional. Se ha utilizado para visualizar los datos guardados a través del chat con el bot. Al iniciar la conversación, se crean varias tablas y luego hay que importar el fichero creado, que es la BBDD, al programa.

Capítulo 6

Arquitectura del sistema

Después de realizar un análisis exhaustivo de los requisitos de nuestro sistema, se concretó la arquitectura del sistema de la siguiente manera:

6.1 Estructura de BBDD

Las tablas necesarias para el bot son:

- andreas: se guardan los usuarios del rol Andrea. Andrea puede rechazar y aceptar varios vídeos.
 - andrea_username (text): el usuario que se usa en el Telegram.
 - videos_rejected (text): el usuario del vídeo que ha rechazado Andrea.
 - videos_accepted (text): el usuario del vídeo que ha denunciado Andrea.
- centers: tiene las localizaciones de los centros para poder validar las identidades.
 - postal_code (INTEGER): el código postal del centro.
 - address (text): la dirección completa del centro.
 - name (text): el nombre del centro.
- redes: los datos de redes.

- red_id (INTEGER): el ID del usuario con el rol Red.
- red_username (text): el usuario del Telegram con el rol Red.
- category (text): la categoría de una Red. Puede ser Verde, Violeta, o Coral.
- vídeo: NULL si no ha subido nada, 1 si ha subido algo.
- photo: NULL si no ha subido nada, 1 si ha subido algo.
- gender: el género de la Red. Es necesario para que luego el usuario de Andrea pueda elegir el género para los vídeos.
- reports: se inserta los datos cuando Andrea denuncia un vídeo de una Red.
 - red_username (text): el usuario de la Red denunciado.
 - andrea_username (text): el usuario de la Andrea que ha denunciado.
- users: se guarda los datos de todos los usuarios.
 - id (INTEGER): el ID de todos los usuarios.
 - username (text): el usuario del Telegram de todos los usuarios.
 - gender (text): el sexo de los usuarios.
 - user_role (text): el rol del usuario. Puede ser Red o Andrea.
 - postal_code (text): el código postal de la ubicación del usuario.
- groups: se guarda los datos de todos los grupos del chat.
 - id (INTEGER): el ID / código postal de cada grupo .
 - enlace (text): el enlace generado para la invitación de grupo.

Capítulo 7

Desarrollo del sistema

A continuación se comentará la implementación del bot, sus funcionalidades y como sería la experiencia de usuario.

7.1 Implementación del bot

Durante la ejecución del bot podremos diferenciar entre dos roles distintos, Red y Andrea, y a su vez dentro del primer grupo, podremos disgregar en tres tipos de redes diferentes: Coral, Violeta y Verde.

Andrea

Andrea esta pensada para usuarias cuya situación personal sea difícil, personas que necesiten ayuda por sufrir violencia machista y quieran cambiar su vida. Para ello, y tras unos pasos iniciales, podrán elegir conversar con un grupo de mujeres en su misma situación o personas incluidas dentro de la Red que estén en un entorno cercano geográficamente hablando.

La idea es que, una vez eliminen la falta de confianza, o mejor dicho, aumente ésta, podrán llevar la relación al siguiente nivel para que, estas personas que comentábamos antes ya sean Andrea o Red, les acompañen en sus tareas diarias, para así reducir el temor de sentirse solas en el día a día.

Red

Pertenecen a la **Red** aquel grupo de personas que quiera participar en la ayuda al grupo Andrea. Para ello, se han marcado una serie de requisitos, aconsejados por las colaboradoras del Bot de la UNED, para aumentar la seguridad y fiabilidad de la herramienta.

Red Verde

Cualquier persona que interactúe con el bot pasará a formar parte de la **Red Verde**, pero no podrá tener contacto con una Andrea hasta que no sea validado por parte de una Red Coral.

Antes de que estas personas sean verificadas deberán enviar un vídeo y una foto del DNI, posteriormente deberán dirigirse a la Asociación de Violencia de Género más cercana para que se pueda confirmar que es la misma persona y que además no es peligrosa.

Las personas sin verificar no aparecerán en la lista de personas que pueden ayudar a una Andrea, de esta forma siempre estarán protegidas ante posibles agresores o personas que simplemente quieran hacer mal uso del bot. Así mismo, con este método de verificación hacemos que sea fácil para cualquier persona colaborar y ayudar, ya que el proceso no es excesivamente tedioso.

Red Violeta

Dentro de la **Red Violeta** se encuentran aquellas personas verificadas por una de las asociaciones de mujeres adscritas al proyecto y por tanto, las que ya pueden interactuar con aquellas usuarias de tipo Andrea.

Red Coral

La **Red Coral** está formada por aquellas usuarias encargadas de la administración del bot, no a nivel técnico si no a nivel de gestión de usuarias. Serán las personas encargadas de promover a una usuaria de la red verde a la red violeta, de crear los diferentes grupos de Telegram en caso de que fuese necesario hacer uno nuevo. Esto último es una limitación de los bots en esta aplicación y por eso es necesaria la intervención de una persona.

7.2 Diagrama de secuencia

A continuación, se muestra el diagrama de flujo del bot [@red_Andrea_bot](#), en el cual podemos ver los pasos que debe seguir cada usuario o usuaria del bot, dependiendo del rol que tenga, y las acciones del bot en función de los botones pulsados o acciones realizadas.

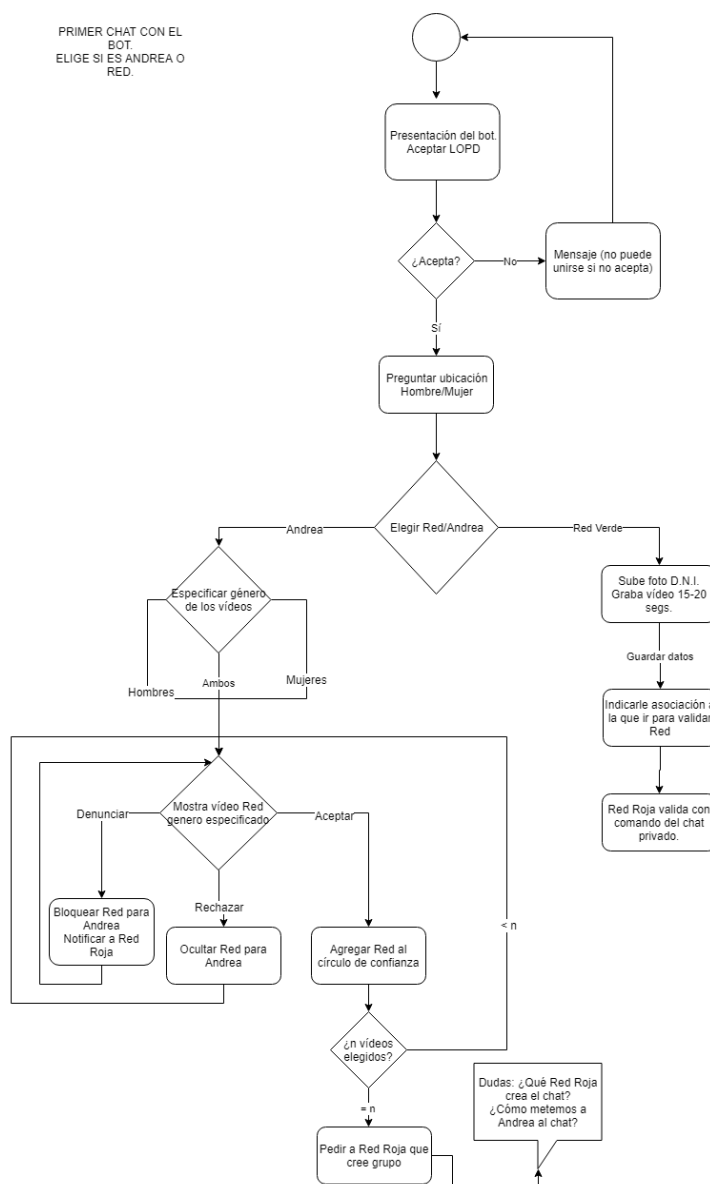


Figura 7.1: Diagrama de flujo del Bot

7.3 Experiencia de usuario - Primer uso

Como podemos apreciar en la imagen 7.2 esto sería lo que podemos esperar del bot. En un primer momento tenemos la opción de [/start](#) para iniciar la conversación con el bot. Nada mas iniciarla recibimos una breve introducción del bot y acto seguido un documento de la ley de protección de datos (LOPD), hay que leer el documento y posteriormente aceptarlo mediante un botón en el teclado (ACEPTAR). En el caso de rechazarlo, se muestra como la imagen 7.3.

El siguiente punto es una valoración de la usuaria, que a su vez es una capa de seguridad para intentar paliar el intrusismo, deberá responder una serie de preguntas para validar que es ella realmente y poder valorar el grado de ayuda que necesita.

Una vez contestadas todas las preguntas se solicitara la ubicación actual. Esto ayudará para poder ofrecer el grupo mas cercano, siempre y cuando la usuaria quiera entrar en algún grupo. En todo momento la usuaria tiene el control del entorno que maneja dentro del bot. Si lo desea, puede borrar todos sus datos y desinstalarlo. Si no se desea enviar tu ubicación exacta por algún motivo, no pasa nada, ya que simplemente se puede indicar el código postal y con ello, se determinará que redes de apoyo tienes cerca.

El siguiente dato a cumplimentar será saber si la persona se trata de un hombre o una mujer, ya que más adelante se utilizará para poder filtrar cuando se precise ayuda o simplemente se desee mantener una conversación, pudiendo elegir entre tu mismo sexo o el contrario.

Una vez contestadas todas las preguntas necesarias para el registro y validar la autenticación de la usuaria, procedemos a explicar las diferentes redes de las que dispone el bot, y cuales son los objetivos para permanecer en ellas.

Andrea

Andrea está pensada para usuarias que necesiten ayuda, conversar con alguien o que les acompañen a cualquier sitio. Pensada para personas que están en riesgo o creen que puedan estarlo.

Una vez elegida la red a la que quieres pertenecer, te saldrán distintas opciones, en este caso como estamos con el rol Andrea 7.4, las opciones son las siguientes.

Se mostrarán una serie de vídeos que subieron usuarios del bot donde ofrecían ayuda (red). Una vez elegido el filtro por sexo (hombre o mujer) se podrá decidir, como se aprecia en la figura 7.4 si aceptar o rechazar a dicho usuario, incluso aparece la opción denunciar.

Mediante estos vídeos se ofrece a las usuarias la opción de familiarizarse un poco más con las personas que vayan a entablar conversación. De esta forma, se facilita que sea lo más accesible para estas el pedir ayuda, mantener una conversación, etc...

Red

El primer paso, como ya se ha comentado en puntos anteriores, será registrarse y rellenar las distintas preguntas que nos haga el bot. Una vez elegida la red, se pedirá que se grabe un vídeo para que las usuarias puedan elegir con quien hablar o pedir ayuda. Se puede decir que sería un vídeo introductorio donde se transmita confianza y seguridad. La Red es **Verde** actualmente.

Enviar el vídeo y la foto es para asegurar que eres la persona quien dices ser en las preguntas. En ningún momento este vídeo será publicado o visionado por otros usuarios sin tu consentimiento. Del mismo modo el vídeo introductorio sirve para que en un futuro y siempre que tu lo apruebes, otros usuarios puedan verlo y elegir si desean hablar contigo o pedirte ayuda. De esta forma es un poco mas cálido y cercano el escribir a alguien.

Una vez enviado el vídeo se requiere la autenticación del DNI, es otra capa de seguridad con la cual se transmite una mayor confianza y seguridad del bot.

Tanto el vídeo como el DNI sera verificado por la asociación más cercana en base a su ubicación. Sera necesario ir personalmente para completar el registro en la red satisfactoriamente.

Una vez finalizado el proceso de registro ahora es Red **Violeta** y se tiene diferentes opciones.

Red violeta se crea para las usuarias que quieran ofrecer ayuda a otras, ya sea para acompañarlas a lugares concretos, médico, ofrecer una conversación. Para poder pertenecer a esta red tendrás que seguir los pasos que podemos apreciar en la imagen siguiente:

Como se ha comentado siempre tiene el control el usuario, y entre otras opciones una de ellas es ROL, la cual permite cambiar de rol cada vez que el usuario lo desee, tiene la opción de ver el vídeo y foto que subió en el registro, opción de ver un listado de asociaciones o lugares de interés, borrar sus datos, por si se quiere "desaparecer" de nuestro sistema, etc....

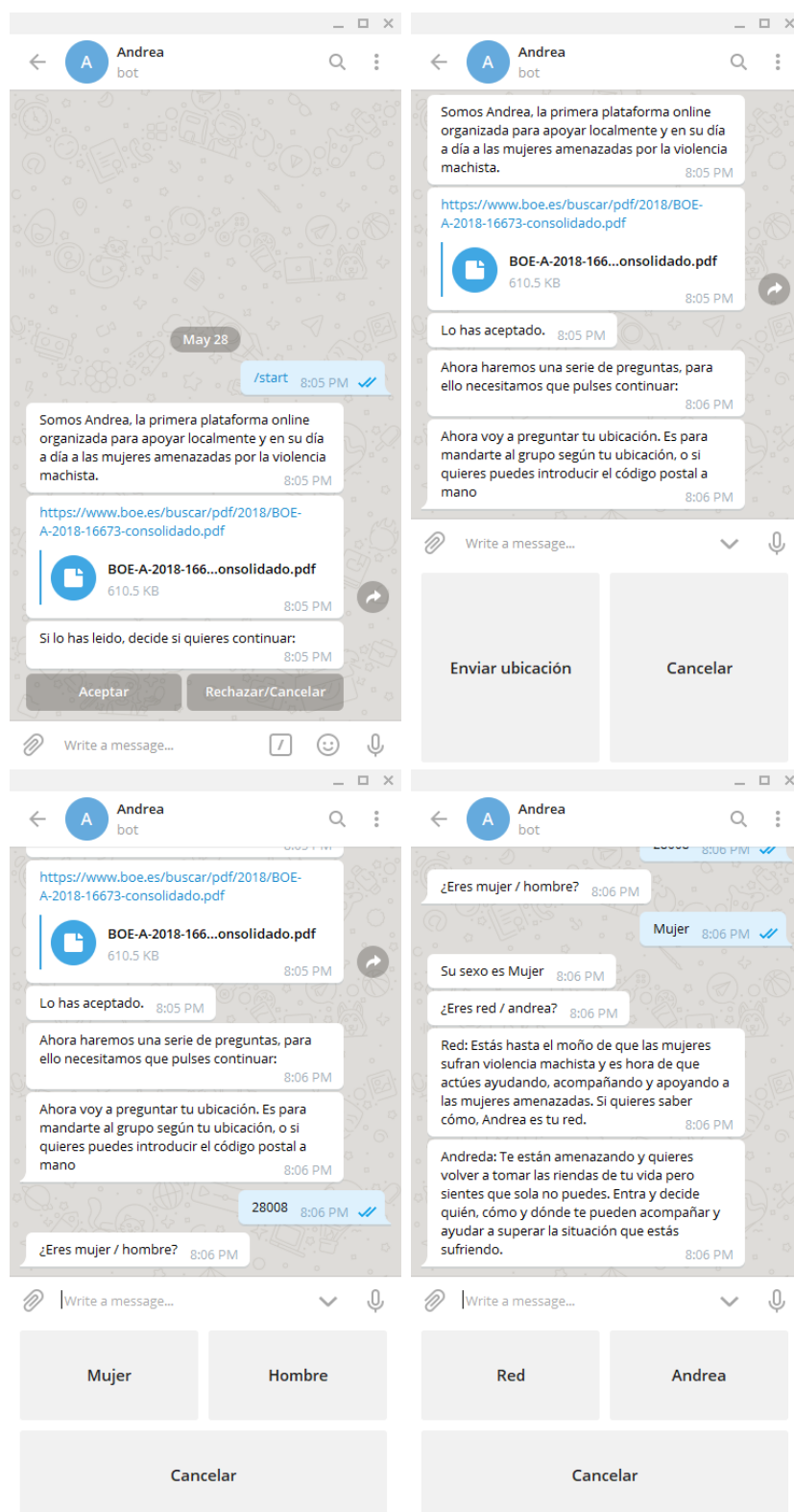


Figura 7.2: Ejemplo de primera conversación con el bot

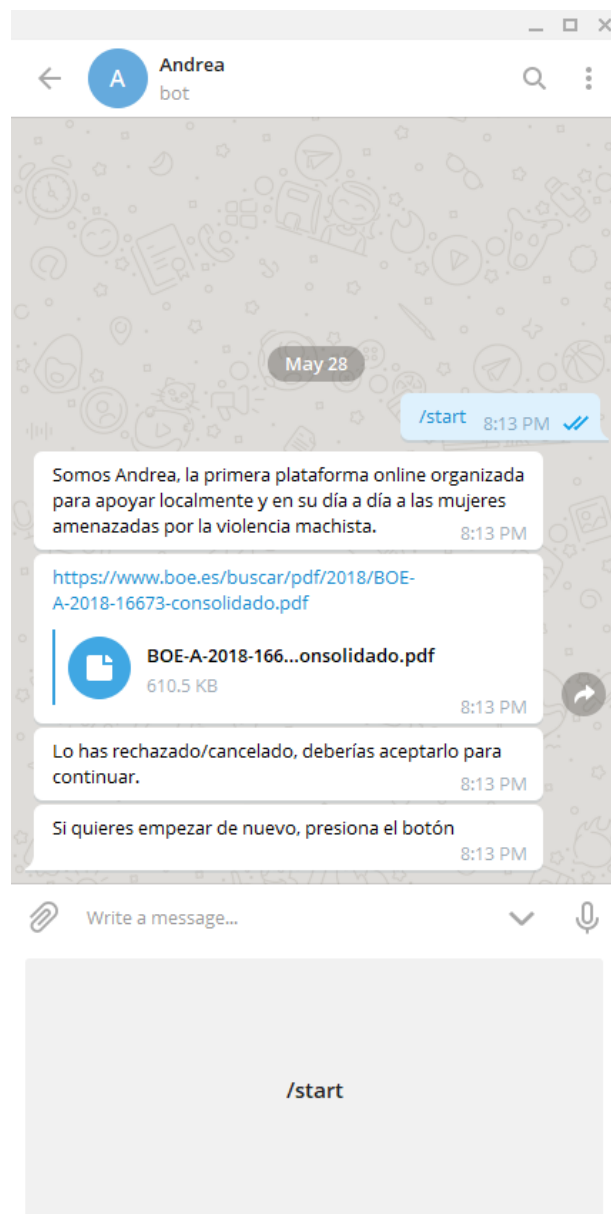


Figura 7.3: El mensaje que muestra el bot en el caso de rechazar el
mento de la LOPD

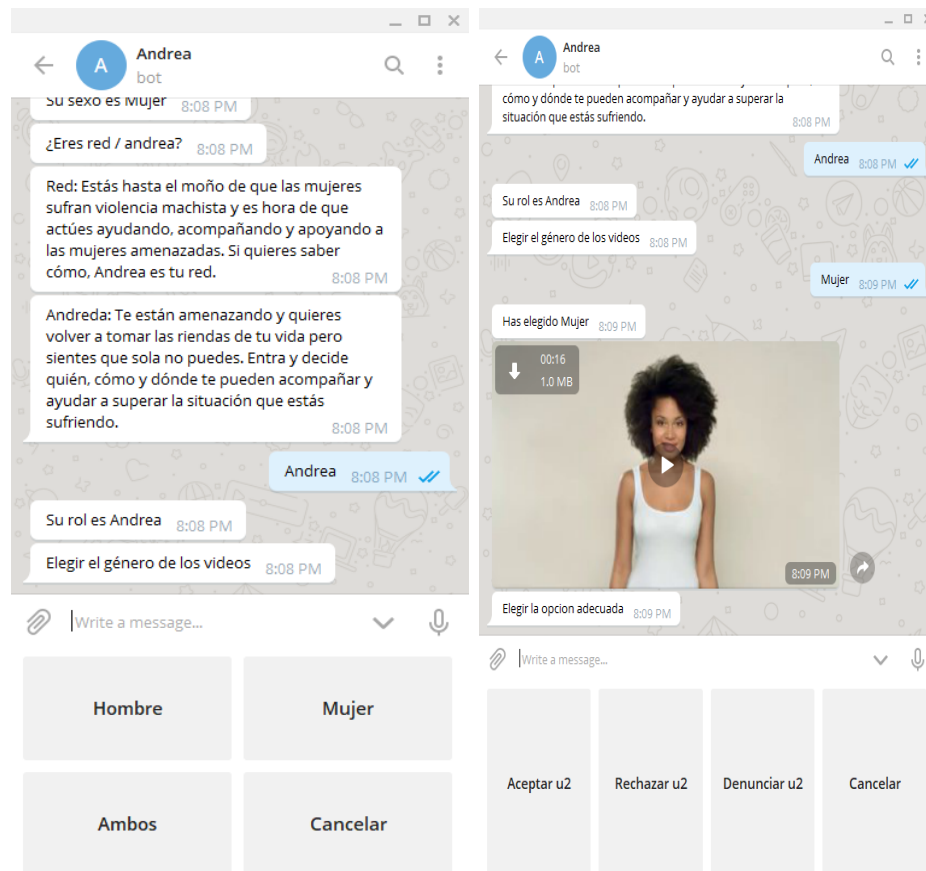


Figura 7.4: Conversación desde el rol Andrea

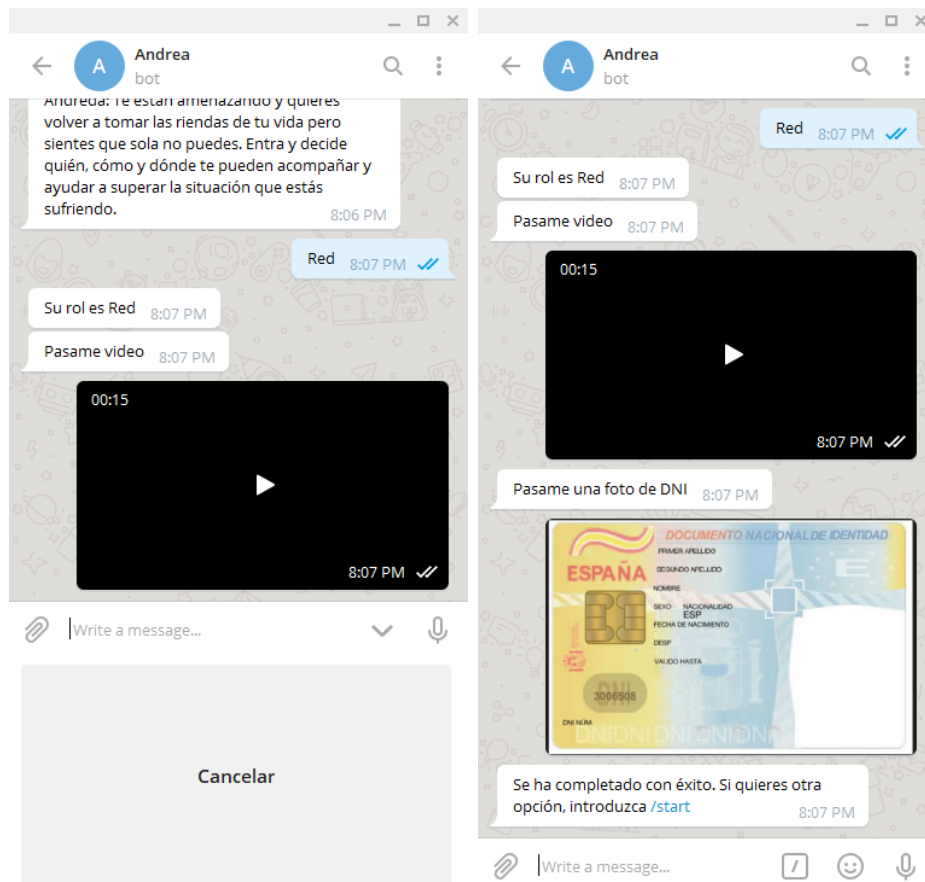


Figura 7.5: Conversación desde el rol red

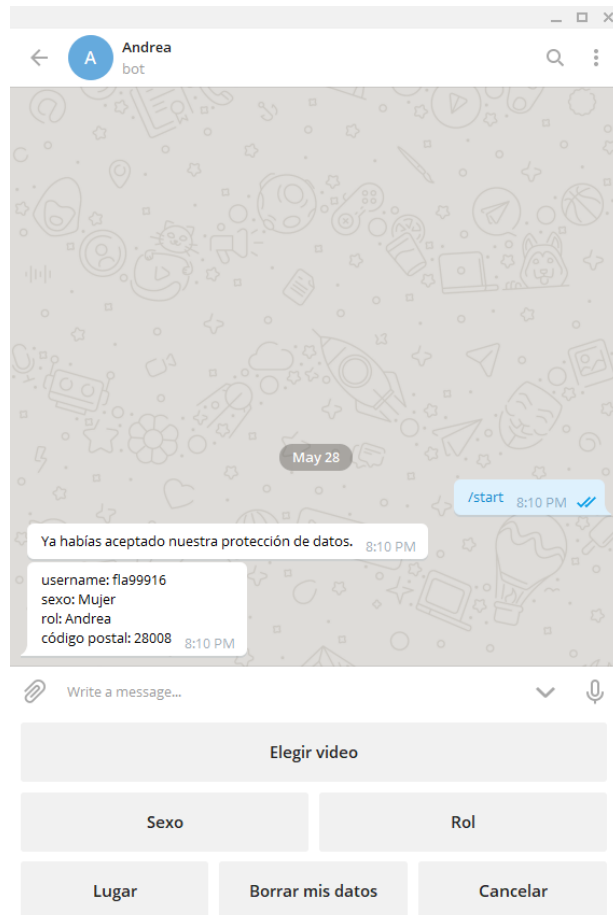


Figura 7.6: Diferentes opciones que permite el bot desde el rol Andrea

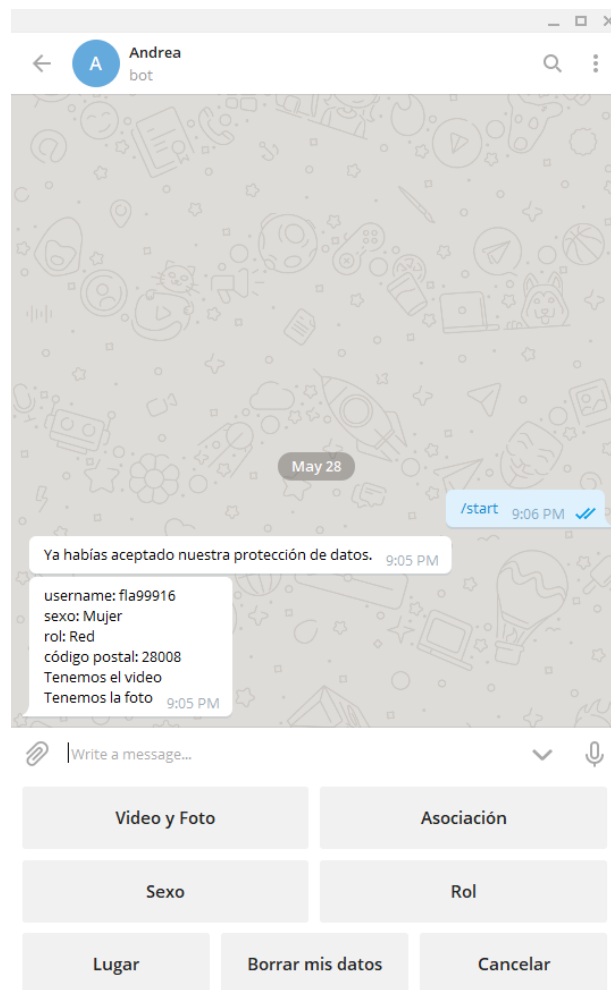


Figura 7.7: Diferentes opciones que permite el bot desde el rol Red

Capítulo 8

Puntos críticos del proyecto

En este apartado, vamos a exponer los puntos en los que los miembros del equipo nos hemos encontrado con mas dificultades.

Requisitos

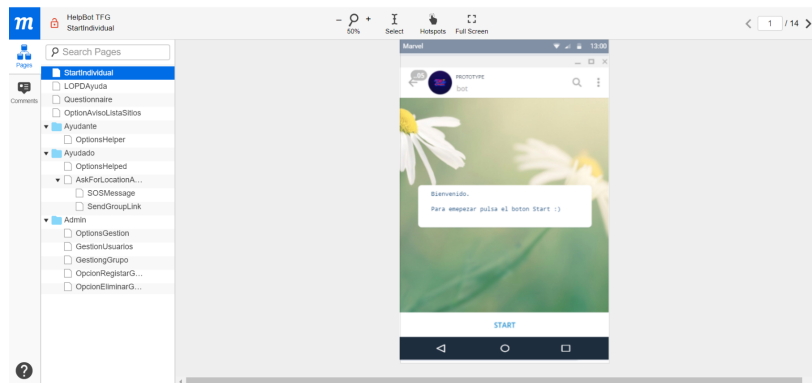


Figura 8.1: Imagen de la herramienta de diseño

Al inicio del TFG tuvimos dudas de como enfocar el proyecto ya que no conocíamos por completo las características y funcionalidades que se tenían pensadas.

Se realizó una pequeña demo utilizando la aplicación web Moqups mediante la cuál pudimos mostrar a los representantes de la UNED nuestra idea de una red de apoyo enfocada a Telegram. De esta manera se

pudo aclarar que ideas eran buenas y que ideas había que repensar de una mejor forma.

A partir de este punto el equipo ya tuvo mucho más claro por donde se podía empezar a desarrollar el bot.

Git

Al ser un equipo de trabajo de tamaño medio, era necesaria la utilización de un repositorio para poder trabajar sobre el mismo proyecto sin problemas de sobreescrituras. Todos los cambios son reflejados en el documento final.

La experiencia del equipo con Git se basaba en pulls de repositorios, pero para trabajar en equipo necesitábamos tener un mayor control sobre todas las funcionalidades que ofrece Git, por lo que durante las primeras semanas nuestro director nos dio acceso a un repositorio ubicado en un servidor de la Universidad Complutense de Madrid para que realizáramos pequeños ejercicios sobre él.

Al acabar este proceso de aprendizaje, ya fuimos capaces de empezar a desarrollar módulos del bot sobre el mismo proyecto.

Aplicación de la Ley Orgánica de Protección de Datos (LOPD)

Esta sección es un punto candente dentro del desarrollo del bot, ya que, toda esta información de carácter personal y según la nueva y vigente ley, hay que tratarla de manera segura. Para ello, la usuaria tiene que poder tener acceso en cualquier momento a qué datos tiene el bot sobre ella y derecho al olvido, es decir, a poder eliminarlos completamente si lo desea.

Lo primero que se planteó respecto a este tema, fue saber que datos iban a ser totalmente necesarios para la ejecución del bot. Una vez este aspecto fue definido en la reunión que se mantuvo con los representantes de la UNED, se decidió enviar, en los primeros pasos dentro del registro en el bot, un documento / “disclaimer” informando sobre lo que se iba a hacer con la información personal que estaban compartiendo. Con ello, y únicamente aceptando los términos y condiciones de uso, se da paso a continuar con el uso de Andrea.

Python

Al igual que con Git, ninguno de los miembros del equipo tenía experiencia con Python, siendo el lenguaje que habíamos elegido para desarrollar el bot. Python se creó con el objetivo de hacer un lenguaje de programación ágil y sencillo, con una curva de aprendizaje muy corta. Encontramos muchos ejemplos iniciales en internet de bots desarrollados con Python, así que fue fácil empezar a crear nuestros bots "hola mundo". Cada miembro del equipo realizó un bot con alguna funcionalidad sencilla, algunos ejemplos son un bot calculadora, que tenía funciones sencillas como calcular una sucesión de Fibonacci o averiguar si un número es par o impar; un bot "adivino" que contestaba Si, No o Quizás de manera aleatoria; un bot mayúscula, que pasaba las palabras que recibía del usuario a mayúsculas

Disminución del número de personas del grupo

Tras un periodo de tiempo transcurrido tras el comienzo del proyecto, y a posteriori de haber fijado levemente los requisitos y la carga de trabajo de cada uno, una de las integrantes del equipo, dejó de contestar a los mensajes del grupo, de venir a las reuniones,... por lo que se dio a entender que había abandonado el trabajo. Esto causó que, toda esa carga que le había sido asignada, tuviéramos que asumirla entre los demás integrantes.

Capítulo 9

Conclusiones del proyecto

A lo largo del desarrollo de este TFG, hemos podido participar en una de las ramas de la informática que mas crece en la actualidad, los bots y la Inteligencia artificial, además de poder ayudar a una causa, desgraciadamente a la orden del día, mediante ello.

Por otro lado, como algunos ya estamos acostumbrados al mundo laboral, somos conscientes de que hay que adaptarse al uso de nuevas herramientas cada día, esto nos ha ocurrido durante todo el desarrollo de Andrea. Hemos tenido que invertir tiempo en el aprendizaje de herramientas y lenguajes de programación, pero esto ha hecho que mejore el proyecto y que mejoren nuestras propias skills.

La propuesta de proyecto inicial nos pareció atractiva ya de por si, por su candencia en el mercado tecnológico actual. A esto se añadió el poder enfocararlo hacia algo con una utilidad real y social como es una red de apoyo local, nos pareció muy interesante poder ayudar a personas en riesgo de violencia con los conocimientos adquiridos durante el grado.

A la hora de comenzar a investigar sobre bots de Telegram nos dimos cuenta de la amplia comunidad que hay detrás de ello, esto nos fue de mucha ayuda para comenzar. También cabe destacar la API de python que nos ayudó a consumir de los servicios de Telegram con mayor facilidad, llamada telegram-python-bot con licencia LGPLv3 de software libre.

Durante los años en el grado hemos adquirido conocimiento y experiencia, pero el desarrollo de este TFG ha sido una gran prueba a nivel colectivo. Todas las pruebas, reuniones, investigación y desarrollo nos han valido para estar preparados para situaciones que vivimos y viviremos en el mundo laboral.

Con el paso de los meses hemos ido viendo como nuestro bot iba cogiendo forma y esperamos y deseamos que tenga una utilidad real para muchos usuarios y usuarias.

Throughout the development of this end-of-degree project, we have been able to participate in one of the most growing branches of computing today, bots and artificial intelligence, as well as being able to help a cause, unfortunately the order of the day, through it.

On the other hand, as some of us are already accustomed to the world of work, we are aware that we must adapt to the use of new tools every day, this has happened to us throughout Andrea's development. We have had to invest time in learning programming tools and languages, but this has made the project better and improve our own skills.

The initial project proposal seemed attractive to us already, due to its presence in the current technological market. To this was added the power to focus on something with a real and social utility as is a local support network, we found it very interesting to be able to help people at risk of violence with the knowledge acquired during the degree. At the time of beginning to investigate about bots of Telegram we realized the wide community that is behind it, this was very helpful to start. Also noteworthy is the python API that helped us consume the Telegram services more easily, called telegram-python-bot with LGPLv3 free software license. Par During the years in the degree we have acquired knowledge and experience, but the development of this TFG has been a great test at a collective level. All the tests, meetings, research and development have enabled us to be prepared for situations that we live and will live in the workplace.

Over the months we have been seeing how our bot was taking shape and we hope and wish that it has a real use for many users.

Capítulo 10

Ampliaciones futuras

En este capítulo se van a describir las funcionalidades que por tiempo y complejidad se han podido desarrollar antes de esta exposición, pero que nos comprometemos a continuar una vez acabado el TFG.

10.1 Botón de pánico

En la reunión que se realizó con las distintas autoridades de la UNED, se comentó en varias ocasiones la inclusión de un botón del pánico dentro del propio chat con Andrea. Este botón borraría los mensajes de conversaciones involucradas con Andrea, compartiría ubicación en directo con los miembros de la red Violeta de la usuaria, y también cabría la posibilidad de que llamara a la policía.

10.2 Puntuación de usuarios

En las reuniones que mantuvimos, se comentó el desarrollo de un sistema de puntuación de cada usuaria, basado en la experiencia del resto de usuarias. Así aumentaríamos la confiabilidad del resto de personas que pertenecieran a la red y, en parte, ayudaríamos en la decisión de seleccionar a aquellas personas de la Red Violeta que se comentaba anteriormente.

10.3 Ampliación de zonas de chats

El bot está destinado para la provincia de Madrid de momento, y tiene posibilidad de ampliar las zonas para el uso del bot ya que la integración de las asociaciones de otras provincias es sencillo. Sería conveniente si creara las tablas de asociaciones por cada provincia en vez de controlarlo en una tabla. Ya que es más sencillo manejar los datos por módulos.

10.4 Inteligencia artificial

En las labores de investigación nos pareció interesante dotar a nuestro bot de la capacidad de mantener conversaciones básicas para ayudar a la usuaria en sus primeros pasos con Andrea.

Por ejemplo, dialogflow, es una API para crear chatbots con reconocimiento de lenguaje natural y machine learning. Además se puede integrar en Telegram y el pricing es gratuito. Proporcionaríamos la tecnología de Machine Learning y reconocimiento del lenguaje natural para que podamos proveer a las usuarias de conversaciones fluidas y naturales con nuestros bots.

Capítulo 11

Aportaciones personales

11.1 Adrián Córdoba Clapés

Conocimientos

Durante estos años de universidad, he adquirido distintos conocimientos relacionados con el mundo de la Informática y más aún, en lo relacionado con el Hardware. Lo primero de todo, el descubrimiento y manejo, en mayor o menor medida, de distintos lenguajes de programación, tanto a bajo nivel, por ejemplo, ensamblador, C, VHDL, etc... como Java, C++, HTML5, de alto nivel e intentando que todos ellos, optimizaran los recursos disponibles en la mayor manera posible. Además de todo ello, también he aprendido a como gestionar un proyecto desde la definición de requisitos hasta el mantenimiento de la aplicación final; a conocer la arquitectura interna de un sistema empotrado, de un ordenador o de un sistema distribuido formado por varios racks, por ejemplo. Por último, destacar que gran parte de todo ello, lo aplico diariamente en mi trabajo actual como analista-programador de Sistemas Operativos de Tarjetas Inteligentes.

Investigación

Para este proyecto, partimos prácticamente desde cero. Tanto a nivel de conocimientos de bots de Telegram, ya que mi manejo con esta aplicación únicamente había sido a nivel de usuario; como a nivel de manejo del lenguaje de programación Python, por lo que, lo primero que tuve que

hacer fue estudiar las características de ambos temas. Para el bot, busqué el API que íbamos a utilizar y algunos ejemplos en internet para ver la estructura que tenían estas herramientas de Telegram. Además, al estar escritas en Python la mayoría de ellos, también me sirvió para aumentar mis conocimientos. Aún así, para reforzarlos, realicé un curso online de este lenguaje.

Una vez sobrepasado este punto y partiendo de una base un poco más firme, comenzamos a desarrollar pequeños bots de prueba, como si de un “Hola mundo” se tratase, para amoldarnos un poco más a todo esto. Posteriormente, intentamos imaginarnos los requisitos que podría tener una herramienta como la que nos solicitaban: un bot que pudiera ayudar a las víctimas de violencia de género. Para ello, buscamos información relacionada con el tema: estadísticas, informes, etc... que pudiera orientarnos un poco más en la definición de requisitos. Una vez aterrizamos estos, desarrollamos otros bots con pequeñas funcionalidades que se adaptaran a estas características que creíamos que se iban a necesitar en el bot, como por ejemplo, un bot que automatizara, en base a la ubicación, el envío de un grupo de Telegram de gente cercana. Gracias a ello descubrimos las posibilidades reales que ofrecía Telegram y pudimos reunirnos, con una buena base de conocimientos, con la gente de la UNED, que eran aquellas personas que nos habían solicitado el desarrollo este proyecto. Para esto, preparamos diagramas de flujos y una breve ejecución con “pantallazos” de como nos imaginábamos que podría ser el bot y junto al resto de gente con la que nos habíamos juntado, decidimos que había que desarrollar finalmente para lanzar el bot y que se podía dejar para un futuro.

Finalmente, hemos llegado a tener un bot que se apoya en Bases de Datos (la cual también investigamos para ver que opción era mejor), utiliza la localización, distingue entre varios roles, organiza grupos, utiliza teclas propias,... En definitiva, interactúa, de una forma aceptable con el usuario en base a los requisitos fijados.

11.2 Carlos Ambrona Palacios

Conocimientos

Como estudiante del Grado de Ingeniería de Computadores he estado adquiriendo durante los cuatro cursos, distintos conocimientos sobre varios lenguajes de programación, como por ejemplo Java, C++, C o VHDL. También he aprendido a realizar proyectos de principio a fin. Realizando un escueto boceto como punto de partida, diseñar la arquitectura del mismo y por último realizar labores de mantenimiento y mejora del proyecto, el cual para el mundo laboral es necesario y sobretodo muy útil. Me han enseñado a enfrentarme a distintos problemas que puedan surgir desarrollando programas o aplicaciones. Saber cual puede estar el error y sobretodo y mas importante, como solucionarlo o como poder buscar información para ser capaz de resolverlo.

Actualmente estoy trabajando como desarrollador de software en entornos Big Data, donde aplico mucho de los conocimientos adquiridos en la carrera y en el TFG.

Investigación

Al ver qué TFG se ofrecían para este curso, vi que uno de ellos estaba relacionado con la aplicación de mensajería instantánea Telegram. Soy usuario desde que se lanzó la aplicación prácticamente, por ello me pareció una buena elección, a parte sabía que otros TFGs sobre la misma aplicación habían acabado utilizando el lenguaje de programación python. Por lo que me decante por este TFG sin duda, ya que auna dos puntos fuertes, uno es la aplicación que uso a diario y el otro es un lenguaje de programación que quiera aprender, ya que es muy demandado en el mercado laboral actual.

Una vez aceptado para formar parte del TFG, empecé a investigar como podríamos desarrollar el bot en Telegram. El primer paso fue ver ejemplos sencillos en internet y ver qué API utilizaban, asi poder comparar con el material de apoyo que nos proporcionó nuestro tutor. Una vez hecho el análisis de la API a implementar tuve que aprender python. Miré varios tutoriales para poder empezar a desarrollar nuestro bot.

El primer bot que realice fue bastante sencillo, ya que solo quería una primera toma de contacto, tanto con la API que íbamos a utilizar como

con el lenguaje python. Su única función era repetir lo que escribías, de manera que tenía un bot básico, que enviaba y recibía mensajes al usuario.

Realizada la labor de investigación y ver qué limitaciones tenía el bot implementado con la API y python nos reunimos con nuestros clientes (profesores de la UNED), a ellos se les ocurrió esta aplicación, expusimos las limitaciones y nos encargamos de desarrollar el bot lo más completo posible.

Además, donde he estado más involucrado ha sido en la lectura de los mensajes recibidos al bot, los cuales guardamos en un JSON para registrar la respuesta de cada usuario a las preguntas de validación, automáticamente se guardan en la BBDD para no volver a preguntar a la usuaria. Además de completar distintas funcionalidades del desarrollo, ayudar a la resolución de errores e implementación de botones.

Por último, mencionar mi aportación en la memoria. Destacar que entre todos los miembros del TFG decidimos utilizar formato LaTeX para realizar la memoria. Una vez decidimos esto, investigamos que editor sería el mas acorde a nuestras necesidades, entre varios elegimos Overleaf, así tendríamos la ultima versión siempre disponible.

Al principio no fue fácil ya que desconocía este formato y sus comandos o reglas. Nuevamente necesité mirarme varios tutoriales donde aparecían tanto comandos básicos como avanzados.

Una vez conseguida una noción básica sobre LaTeX comenzamos creando el índice y los capítulos/secciones que debería tener la memoria como base.

Con la estructura básica hecha nos repartimos el trabajo muy dinámicamente, ya que cada vez que accedíamos a rellenar la memoria, revisábamos los últimos avances y continuábamos donde lo dejó el último miembro que hizo avances.

De esta manera todos revisábamos lo de todos, para poder reducir los posibles errores de lenguaje, evitar desarrollar temas similares en distintos capítulos o apartados, asegurarnos de que hemos completado bien cada capítulo y apartado y tener una idea de lo mas completa de la memoria.

11.3 Carlos Manuel Perdomo Fernández

Conocimientos

Como estudiante del grado de ingeniería de computadores, he adquirido conocimientos de programación en distintos lenguajes y manejo de base de datos.

En mi carrera profesional, me dedico al sector de las aplicaciones web.

Investigación

Al descubrir sobre la oferta de este TFG, mi interés fue total ya que conozco Telegram prácticamente desde sus inicios y soy usuario de canales, grupos y bots. Siempre me había llamado la atención en especial la funcionalidad de los bots, por lo que solicité este proyecto como TFG. Nuestro director nos proporcionó los bots de TFG de años anteriores para que pudiéramos ir çacharreando con los nuestros propios. La mayoría de ellos utilizaban APIs en desuso como Telepot.

Investigando, vimos que la mejor opción era la API telegram-python-bot, la cual, es muy sencilla de instalar con tan solo ejecutar el comando "pip install python-telegram-bot.^{en} la consola de python.

Mediante ejemplos de conexión de la API pudimos usar sus funcionalidades básicas en un corto periodo de aprendizaje y usarla en nuestro bot. La conexión al bot se realiza fácilmente usando la clase Updater de la api:

```
üpdater = Updater('TOKEN DE NUESTRO BOT')
```

También estuve involucrado con la elección de la mejor base de datos para un bot de Telegram. Descartamos las bases de datos no relacionales (MongoDB) por no tener ninguno de los miembros del equipo demasiada experiencia con ellas. Finalmente nos decantamos por SQLite por ser la base de datos que mejor se adaptaba a nuestro proyecto, así como por su fácil integración en python.

Como todo motor de base de datos SQL autónomo, puede manejar todo tipo de datos de una manera relativamente simple, todo esto se debe a que se integra directamente con la aplicación en lugar de tener que instalarla en algún espacio y luego conectarla con la aplicación.

Otra de las características de SQLite que nos hizo decantarnos por ella fue su óptimo rendimiento, esto se debe a que todas las operaciones de lectura

y escritura son muy rápidas para la base de datos SQLite, en donde es casi un 35 % más rápido que el sistema de archivos. Otro factor es que solo carga los datos que se necesitan, en lugar de leer todo el archivo y mantenerlo en la memoria, así como también si se edita una parte pequeña, solo sobrescribirá las partes del archivo que se modificó y no en su totalidad.

Otra ventaja es que SQLite es totalmente accesible a través de una amplia variedad de herramientas de terceros como por ejemplo DB Browser, la cual hemos utilizado en el desarrollo de nuestro bot.

Cabe destacar también la parte de investigación en esta memoria, decidiendo crearla con \LaTeX por su estabilidad y su capacidad multiplataforma.

A través de distintas clases de documento y de su conjunto de macros, LaTeX posibilita escribir textos dividiéndolos en capítulos, secciones, subsecciones, controlando en todo momento la numeración y las referencias cruzadas. Construye índices de contenidos, tablas o figuras. Ajusta los tamaños y tipos de letras según la parte del documento en que se hallen. Otro punto a tener en cuenta en mi investigación relacionado con \LaTeX es Overleaf.

Overleaf es un editor colaborativo que publica en línea utilizando \LaTeX . Conocimos esta herramienta buscando en internet como sincronizar mediante un repositorio git un fichero \LaTeX .

Entonces encontramos esta alternativa, la cual nos ha facilitado muchísimo la realización de la memoria, ya que nos ha dado la posibilidad de editar a todos los miembros del equipo sobre el mismo fichero .tex pudiendo ver el resultado al momento, también ofrece la opción de ver en que lugar del fichero esta cada usuario, así como el chat que incorpora la aplicación para facilitar la comunicación. Y otro punto muy importante, es gratis.

El único inconveniente es que cuando compilas el código, también se refresca el resultado final a los compañeros, lo cual puede llegar a confundir, aún así, Overleaf nos ha dado muchas facilidades.

11.4 Marta Huertas Smolis

Conocimientos

Cuando vi la lista de proyectos propuestos me llamó la atención este proyecto por ser un bot de Telegram, ya que es algo que siempre he querido aprender pero nunca había tenido ocasión para hacerlo.

Una vez había solicitado formar parte del equipo, Carlos, nuestro director del Trabajo de Fin de Grado, me envió toda la información sobre el propósito de dicho bot, lo cual me resultó muy interesante a la vez que útil.

Al poco tiempo de comenzar el curso tuvimos una reunión todos los miembros del equipo junto con el director, en dicha reunión cada uno expuso su situación y disponibilidad, así como su opinión sobre el proyecto que íbamos a llevar a cabo.

Tras esta reunión me empecé a plantear qué funciones podrían ser útiles para una persona que sufre violencia machista y quiere utilizar nuestro bot, así junto con las ideas de los demás compañeros y las funcionalidades que ya habían solicitado los responsables del proyecto nos dispusimos a analizar y plantear el diseño del bot, para a continuación buscar información y estudiar la viabilidad de desarrollar dichas funciones.

Durante el desarrollo de este trabajo he ampliado mis conocimientos de Python, ya que tenía una base gracias a una asignatura optativa pero no era suficiente para el desarrollo de un bot de Telegram como este. Además, he aprendido más sobre la integración con bases de datos y el uso de librerías, como ha sido Telegram-Python-Bot en este caso.

Otra parte importante ha sido la de la gestión del proyecto y la organización de todos los miembros, ya que a pesar de haber trabajado en grupo a lo largo de la carrera, en este caso hemos tenido que hacer un esfuerzo extra para analizar los requisitos y, entre todos, decidir cuáles eran las mejores opciones para poder cumplirlos.

Sim embargo, lo que más me ha marcado en este proyecto ha sido descubrir la realidad de las mujeres que sufren violencia machista, ya que es difícil ponerse en una situación que no has vivido. Personalmente creo que tras este proyecto todos tendremos más empatía hacia las personas que sufren violencia, especialmente las mujeres, y tendremos en cuenta todos los estragos psicológicos que esto causa. Este punto es importante porque es necesario comprender a la otra persona para poder ayudarla.

Por otro lado, me llevo una muy buena experiencia tras haber tenido la reunión con los profesores y expertos de la UNED, he podido ver cómo piensan, los diferentes puntos de vista respecto a este problema y las diferentes ideas para abordarlo.

Investigación

Una vez que tuvimos la primera reunión todos los compañeros y el director, el primer paso fue analizar todo lo que las personas impulsoras de este proyecto querían que cubriese nuestro bot, para dicho análisis tuvimos que investigar a fondo las APIs de Telegram, meditar sobre la manera de utilizar dichas APIs para el desarrollo de las funcionalidades requeridas, y en muchos casos buscar una alternativa para compensar las limitaciones de la herramienta.

Durante el proceso de investigación encontré muchas páginas, foros y repositorios de Github con ejemplos que nos resultaron útiles para la implementación de nuestro bot. Entre ellos se encuentra un bot para compartir paraguas [1], éste bot crea una red de personas que se encuentran en la misma zona en un momento dado, si comienza a llover y alguien no tiene paraguas puede solicitarlo en la aplicación enviando su ubicación para que una persona con paraguas acuda en su ayuda.

Dentro de las funcionalidades que deseábamos añadir, pero que resultaron no ser viables, estaban la posibilidad de añadir y eliminar usuarios de un grupo, así como la creación automática de grupos. En estos casos deberán ser los propios usuarios de la red los que gestionen los grupos, lo cual dependerá de sus roles.

En algunas ocasiones ha sido difícil hallar una respuesta ya que la documentación no estaba clara o no era fácilmente accesible.

11.5 Yelim Kim

Conocimientos

He aprendido mucho de Python, lenguaje que nunca había utilizado y siempre había querido aprender. Es un lenguaje que se utiliza cada vez más debido a su simplicidad y rapidez, elegancia y flexibilidad. Me daba miedo hacer un proyecto en un lenguaje que desconocía pero, al final, he

podido aprender por mi cuenta y gracias a ello conseguimos que el bot funcionara. Me gustaría ampliar mis conocimientos sobre este lenguaje de programación en otros proyectos.

Para la base de datos utilizamos SQLite que es parecido a MySQL, con una sintaxis un poco diferente. Al ser similar a lo que he aprendido en el grado no me ha costado mucho entenderla, salvo la parte de concurrencia (hilos) que me costó un poco más.

Anteriormente había utilizado Git pero, solo para mis proyectos y no había tenido que generar clave ssh para acceder al repositorio, por lo que para el proyecto sí que tuve que buscar en Internet cómo hacerlo. Como equipo inicialmente decidimos utilizar Git, pero finalmente decidimos utilizar Google drive porque teníamos mayores conocimientos. Fue una suerte que decidiésemos cambiar el sistema de control de versiones ya que con esto pudimos evitar la pérdida de tiempo intentando aprender el funcionamiento de Git.

También he aprendido hacer un proyecto entero con un buen equipo. Por ejemplo, tuvimos que afrontar la marcha de un miembro del equipo, debido a ello tuvimos que reorganizar parte del trabajo que nos quedaba por hacer.

Construir el proyecto desde cero fue un desafío. Siempre he trabajado con las plantillas que ofrecían los profesores y seguir sus indicaciones. Sin embargo, esta vez tuvimos que planificar (tiempo de desarrollo de las funcionalidades) y buscar toda la información necesaria para el desarrollo del proyecto (lenguaje de programación, sistema de control de versiones, librerías e información sobre protección de datos entre otras cosas).

Investigación

Me llamó la atención al ver este proyecto en la página web de la Universidad Complutense de Madrid porque el proyecto consistía en hacer un bot de Telegram, ya que utilizo Telegram y diferentes bots y canales todos los días. Además, la intención del bot era interesante, ya que es un problema actual; La violencia de género es un problema que lamentablemente escuchamos a menudo en nuestro país y en el mundo. Con este proyecto vi la oportunidad de aportar mi ayuda para atajar esta situación.

Tuvimos que buscar primero las funcionalidades que nos permitía implementar el bot de Telegram y buscar soluciones ante las restricciones

de la API. A partir de ese punto empezamos a concretar las funcionalidades del bot.

Después de tener la reunión con los expertos de la Universidad Nacional de Educación a Distancia nos dimos cuenta de que anteriormente solo estábamos enfocando las funcionalidades a nivel técnico, es decir, no teníamos en cuenta las funcionalidades necesarias para las víctimas. Por lo que tras la reunión aprendimos la realidad de este problema y las funcionalidades que necesitan y ayudarán a las víctimas.

Deseo que nuestro bot sirva como una ayuda para las mujeres que han sufrido violencia de género.

Bibliografía

- [1] Free Code Camp. *I built a serverless Telegram bot over the weekend. Here's what I learned.* URL: <https://medium.freecodecamp.org/how-to-build-a-serverless-telegram-bot-227f842f4706>.
- [2] Luis Miguel Gilpérez. *El móvil supera por primera vez al ordenador para acceder a Internet.* URL: <https://www.elmundo.es/sociedad/2016/04/04/57026219e2704e90048b465e.html>.
- [3] Google. *Google Drive.* URL: https://www.google.com/intl/es_ALL/drive/.
- [4] IntelliJ Idea. *IntelliJ Idea.* URL: <https://www.jetbrains.com/idea/>.
- [5] El Androide Libre. *WhatsApp también tiene bots, éste se llama qeuBot.* URL: <https://elandroidelibre.elespanol.com/2017/01/bot-para-whatsapp-qeubot.html>.
- [6] Overleaf. *Overleaf.* URL: <https://es.overleaf.com>.
- [7] Telegram Developer Team. *API Telegram Bot.* URL: <https://core.telegram.org/bots/api>.
- [8] Wikipedia. *Anaconda.* URL: [https://es.wikipedia.org/wiki/Anaconda_\(distribuci%C3%B3_de_Python\)](https://es.wikipedia.org/wiki/Anaconda_(distribuci%C3%B3_de_Python)).
- [9] Wikipedia. *Facebook Messenger.* URL: https://es.wikipedia.org/wiki/Facebook_Messenger.
- [10] Wikipedia. *IntelliJ Idea.* URL: https://es.wikipedia.org/wiki/IntelliJ_IDEA.
- [11] Wikipedia. *KIK Messenger.* URL: <https://es.wikipedia.org/wiki/Kik>.

- [12] Wikipedia. *Visual Studio Code*. URL:
https://es.wikipedia.org/wiki/Visual_Studio_Code.